# **Date and Time**

Darbas su datomis ir laiku

## **Sukurti datos objekta**

Sukurti nauja datos objektą galime pašaukus funkcija:

| const currentDate = new Date(); *// return date object of current date and time* |
| --- |

Sukūrus datos objektą ir jį manipuliuodami, mes galime gauti dabartinį laiką ir datą. Kaip pvz. currentDate.getHours(); *// Grąžins datos objekto sukūrimo momento valandas.*

Dabartinis laikas bus kuriamas jūsų kompiuterio ar sistemos lokalus laikas. Jei norite universalaus laiko UTC, tuomet kuriant date objektą reikia konkrečiai specifikuoti.

Jei norime sukurti datos objektą specifiniam momentui, tuomet šaukiant date konstruktorių, galime nurodyti atitinkamus argumentus. Visi galimi argumentai pateikiami žemiau:

| new Date(); new Date(value); new Date(dateString); new Date(year, monthIndex); new Date(year, monthIndex, day); new Date(year, monthIndex, day, hours); new Date(year, monthIndex, day, hours, minutes); new Date(year, monthIndex, day, hours, minutes, seconds); new Date(year, monthIndex, day, hours, minutes, seconds, milliseconds); |
| --- |

**Be parametrų** - šaukiamas konstruktorius gražina dabartinio laiko objektą pagal jūsų kompiuterį ar sistemos kurioje leidžiamas kodas. Jeigu šis objektas šaukiamas iš skirtingų laiko juostų, reikšmė gali skirtis. Arba jei kodas šaukiamas serveryje, serverio ir naršyklės laikas taipogi gali skirtis. Tad į tai reikia atsižvelgti rašant laiko gavimo ir skaičiavimo logiką.

**Value** - tai yra skaitinė datos išraiška pvz: 749019369738. Ši skaitinė išraiška yra milisekundės skaičiuojamos nuo 1970 metų sausio 01 d. UTC (Coordinated Universal Time). Tokiu formatu yra daug lengviau perduoti laiką tarp skirtingų operacinių sistemų, duomenų bazių ar programavimų kalbų.

Pvz:

| const date1 = new Date(0); *// Thu Jan 01 1970 05:30:00* const date2 = new Date(1000); *// Thu Jan 01 1970 05:30:01* const date3 = new Date(5000); *// Thu Jan 01 1970 05:30:05* |
| --- |

**DateString** - tai yra tekstinė datos reikšmė pvz: 'Sun Apr 30 2020 17:27:21 GMT+0300 (Eastern European Summer Time)'. Tokį datos tekstą galima gauti pašaukus Date() funkcija. dateString reikšmių formatų yra įvairių ir šis parametras yra apdorojamas naudojant Date.parse()funkcija.

Date string pavyzdžiai kurie suformatuoji pašaukus atitinkama to string funkcija:

| "Wed Jan 23 2019 17:23:42 GMT+0800 (Singapore Standard Time)" "Wed Jan 23 2019" "23/01/2019, 17:23:42" "23/01/2019" "Wed, 23 Jan 2019 09:23:42 GMT" "Wed, 23 Jan 2019 09:23:42 GMT" "2019-01-23T09:23:42.079Z" "3 march 2015" "3 February, 2015" "2015 3 February" "3 2015 February" "February 3 2015 "February 2015 3" "2 3 2015" "3 march 2015 20:21:44" "3rd February, 2015" // toks formatas negalimas |
| --- |

Pvz:

| const date = new Date("2021-02-10"); |
| --- |

**year** = Tai skaičius reprezentuojantis metus. Skaičiai nuo 0 iki 99 reprezentuoja metus nuo 1900 iki 1999 metų, bet galima vesti ir pinus 1976.

**monthIndex** - Tai skai2ius reprezentuojantis mėnesio indeksą kuris prasideda nuo 0 ir tęsiasi iki

11.

**day** - Tai skai2ius reprezentuojantis mėnesio diena 1 -31.

**hours** - skaičiai reprezentuojantys valandas 0 - 23.

**minutes** - skaičiai reprezentuojantys minutes. 0 - 59.

**seconds** - skaičiai reprezentuojantys sekundes 0 - 59.

**milliseconds** - skaičiai reprezentuojantys milisekundes 0 - 999.

Pvz:

| new Date(2017, 3, 22, 5, 23, 50)  *// Year: 2017,* *// Month: April nes mėnesiai startuoja nuo 0* *// Date: 22* *// Hours: 05* *// Minutes: 23* *// Seconds: 50* |
| --- |

Sukurti data UTC formatu:

| *// 11th June 2019, 12am, UTC.* new Date(Date.UTC(2019, 5, 11)) |
| --- |

Javascript palaiko ISO 8601 datos formata - YYYY-MM-DDTHH:mm:ss.sssZ

| const dt = new Date('2015-02-10T10:12:50.5000z'); |
| --- |

Datos konvertavimas į skirtingus formatus:

| toString() => "Wed Jan 23 2019 17:23:42 GMT+0800 (Singapore Standard Time)" toDateString() => "Wed Jan 23 2019" toLocaleString() => "23/01/2019, 17:23:42" toLocaleDateString() => "23/01/2019" toGMTString() => "Wed, 23 Jan 2019 09:23:42 GMT" toUTCString() => "Wed, 23 Jan 2019 09:23:42 GMT" toISOString() => "2019-01-23T09:23:42.079Z" |
| --- |

Datos Datos generavimas į savo formata

| *// Sugeneruoti datos string formatą* const date = new Date('4-1-2015'); *// M-D-YYYY*  const d = date.getDate(); const m = date.getMonth() + 1; const y = date.getFullYear();  const dateString = (d <= 9 ? '0' + d : d) + '-' + (m <= 9 ? '0' + m : m) + '-' + y;  console.log(dateString); |
| --- |

## **Lyginti datas**

Pvz:

| const date1 = new Date('4-1-2015'); const date2 = new Date('4-2-2015');  if (date1 > date2)  alert(date1 + ' is greater than ' + date2); else (date1 < date2 )  alert(date1 + ' is less than ' + date2); |
| --- |

Daugiau informacijos apie datas:

<https://css-tricks.com/everything-you-need-to-know-about-date-in-javascript/>

## **Date objekto metodai ENG**

| Method | Description |
| --- | --- |
| getDate() | Returns numeric day (1 - 31) of the specified date. |
| getDay() | Returns the day of the week (0 - 6) for the specified date. |
| getFullYear() | Returns four digit year of the specified date. |
| getHours() | Returns the hour (0 - 23) in the specified date. |
| getMilliseconds() | Returns the milliseconds (0 - 999) in the specified date. |
| getMinutes() | Returns the minutes (0 - 59) in the specified date. |
| getMonth() | Returns the month (0 - 11) in the specified date. |
| getSeconds() | Returns the seconds (0 - 59) in the specified date. |
| getTime() | Returns the milliseconds as number since January 1, 1970, 00:00:00 UTC. |
| getTimezoneOffset() | Returns the time zone offset in minutes for the current locale. |
| getUTCDate() | Returns the day (1 - 31) of the month of the specified date as per UTC time zone. |
| getUTCDay() | Returns the day (0 - 6) of the week of the specified date as per UTC timezone. |
| getUTCFullYear() | Returns the four digits year of the specified date as per UTC time zone. |
| getUTCHours() | Returns the hours (0 - 23) of the specified date as per UTC time zone. |
| getUTCMilliseconds() | Returns the milliseconds (0 - 999) of the specified date as per UTC time zone. |
| getUTCMinutes() | Returns the minutes (0 - 59) of the specified date as per UTC time zone. |
| getUTCMonth() | Returns the month (0 - 11) of the specified date as per UTC time zone. |
| getUTCSeconds() | Returns the seconds (0 - 59) of the specified date as per UTC time zone. |
| getYear() | Returns the no of years of the specified date since 1990. *This method is Deprecated* |

| Method | Description |
| --- | --- |
| setDate() | Sets the day as number in the date object. |
| setFullYear() | Sets the four digit full year as number in the date object. Optionally set month and date. |
| setHours() | Sets the hours as number in the date object. Optionally set minutes, seconds and milliseconds. |
| setMilliseconds() | Sets the milliseconds as number in the date object. |
| setMinutes() | Sets the minutes as number in the date object. Optionally set seconds & milliseconds. |
| setMonth() | Sets the month as number in the date object. Optionally set date. |
| setSeconds() | Sets the seconds as number in the date object. Optionally set milliseconds. |
| setTime() | Sets the time as number in the Date object since January 1, 1970, 00:00:00 UTC. |
| setUTCDate() | Sets the day in the date object as per UTC time zone. |
| setUTCFullYear() | Sets the full year in the date object as per UTC time zone |
| setUTCHours() | Sets the hour in the date object as per UTC time zone |
| setUTCMilliseconds() | Sets the milliseconds in the date object as per UTC time zone |
| setUTCMinutes() | Sets the minutes in the date object as per UTC time zone |
| setUTCMonth() | Sets the month in the date object as per UTC time zone |
| setUTCSeconds() | Sets the seconds in the date object as per UTC time zone |
| setYear() | Sets the year in the date object. *This method is Deprecated* |

| toDateString() | Returns the date segment from the specified date, excludes time. |
| --- | --- |
| toGMTString() | Returns a date string in GMT time zone. |
| toLocaleDateString() | Returns the date segment of the specified date using the current locale. |
| toLocaleFormat() | Returns a date string in default format. |
| toLocaleString() | Returns a date string using a current locale format. |
| toLocaleTimeString() | Returns the time segment of the specified Date as a string. |
| toString() | Returns a string for the specified Date object. |
| toTimeString() | Returns the time segment as a string from the specified date object. |
| toUTCString() | Returns a string as per UTC time zone. |
| valueOf() | Returns the primitive value of a Date object. |